

## **Cache attacks against secrete key cryptosystems and Effectiveness of countermeasures**

### **Introduction**

Cryptography came about from a need to secure information. As the field quickly grew, so did the security of the algorithms that where used. As flaws where found in algorithms, new ones would be designed to avoid the attack methods. As these methods of attack where discovered, and understandings of how to write secure algorithms grew, algorithms would remain secure for longer periods of time. Today the field of cryptography has matured to the point where new algorithms are very complex, and hardened against many known attack methods. Many algorithms are analyzed by experts in the filed, and the mathematics behind the algorithm are very well understood and tested. As the algorithm it self becomes less of an available source of weakness, many researchers have begun to look at other methods of attack. Among these other methods, is a category know as “side channel attacks.”

Side channel attacks have not necessarily been considered in most implementations today, and therefore provide a possibility of breaking an algorithm much more quickly than mathematical analysis. As the understanding of these new attacks progress, so will the defenses against them. As with any new method of attack, it is important fully understand these new attack methods, so that with time the defense to them can mature to the point where they do not provide the path of least resistance for attackers.

One such proposed attack involves using cache access of the processor. Such an attack is based not on a direct mathematical fault of the algorithm, rather inadvertent information leakage due to implementation. This gives the attack the advantage of working against more than one specific algorithm, as long as they contain some of the same optimizations. With this form of attack the implementation of the algorithm could cause more problems than the algorithm its self. With a greater understanding of this attack, it should be possible to evaluate the effectiveness of proposed defense against it.

### **Implementation comparisons**

It is most important to find what algorithms and libraries might contain such implementations, as to allow for this attack. The following lists may be considered, depending on available material. The time required to recreate this attack, and then test it for multiple implementations would most likely be far too great for the available time. As such, comparisons will be made based on available research material, and by reviewing specifications and implementations.

#### Algorithms which may be considered

- DES/3DES
- AES
- Twofish
- Serpent
- Blowfish
- RC4 & RC6

#### Libraries which may be considered

- OpenSSL
- Crypto++

## Questions to investigate

While researching this type of attack, I will attempt to find answers for the following questions.

Questions:

- Are the above suggested algorithms/libraries susceptible to these types of attacks?
- Are any of these attacks plausible in a real world environment?
- Defense against these attacks
  - Are there proposed defenses for different types of cache attacks?
  - At what level should defenses be implemented
    - Microprocessor
    - operating system
    - application

As this topic is analytical in nature, verification through testing may not be possible in the given time frame. This should provide a good basis for further expansion in the topic, providing direction to which areas real world test would be most beneficial.

## Project Schedule

March 24 <sup>th</sup> (1 month)	First Progress report: <ul style="list-style-type: none"> <li>• Make list of cache attacks to analyze</li> <li>• List of any proposed defenses to these attacks</li> <li>• Analysis of practicality of the attack in real world (for certain algorithms)</li> <li>• Analytical analysis of atleast ½ the suggested algorithms.</li> </ul>
April 14 <sup>th</sup> (3 weeks)	Second Progress report: <ul style="list-style-type: none"> <li>• Analysis of remaining algorithms/libraries.</li> <li>• analysis of effectiveness of proposed defense</li> <li>• outline the successfulness or need for other defensive measures</li> </ul>
April 28 <sup>th</sup> (2 weeks)	Final progress report: <ul style="list-style-type: none"> <li>• provide draft viewgraphs of presentation</li> <li>• Draw conclusions from previous analysis, determine practicality of these attacks/defenses</li> </ul>
May 8 <sup>th</sup> (1 ½ weeks)	Turn in project report
May 10 <sup>th</sup> (2 days)	Turn in project report reviews
May 12 <sup>th</sup> (2 days)	Final report and Presentation: <ul style="list-style-type: none"> <li>• Provide final report with corrections from reviews</li> <li>• Provide final presentation slides, and present findings</li> </ul>

## Areas of possible change

As stated before, this project involves comparing multiple algorithms and libraries. Due to time constraints, it will not be possible to setup a lab scenario and test all possible combinations for the success or failure of this attack. Likewise it will not be possible to test all the defenses proposed, for all the algorithms and libraries. Therefore any of the above areas might be altered according to the availability of research and literature on the topics. If specifications do not have enough documentation to draw conclusions from, they might not be included in the final analysis. Likewise if other algorithms or libraries are discovered to provide sufficient documentation for analysis, they will be added to the lists above and included in the final report.

## Tentative Table of Contents for Final Report

- Introduction
- Attack explanation
  - Description of specific proposed attacks
- Analysis of secret key ciphers with respect to these attacks
- Analysis of libraries with respect to the attack
- Description of proposed defenses
  - How each might be implemented
  - Discussion of what level defenses are implemented
- Conclusions

## Initial list of literature for this report

- [1] Daniel J. Bernstein, "Cache-timing attacks on AES", November 12, 2004. <http://cr.yp.to/antiforgery/cachetiming-20050414.pdf>
- [2] Daniel J. Bernstein, "Cache-timing attacks on AES", April 14, 2005. <http://cr.yp.to/antiforgery/cachetiming-20050414.pdf>
- [3] Robert G. Salembier, "Analysis of Cache Timing Attacks against AES", Manuscript received May 12, 2006. [http://ece.gmu.edu/courses/ECE746/project/F06\\_Project\\_resources/Salembier\\_Cache\\_Timing\\_Attack.pdf](http://ece.gmu.edu/courses/ECE746/project/F06_Project_resources/Salembier_Cache_Timing_Attack.pdf)
- [4] Dag Arne Osvik, Adi Shamir and Eran Tromer, "Cache Attacks and Countermeasures: the Case of AES" [Extended Version], Revised November 20, 2005.
- [5] Dag Arne Osvik, Adi Shamir and Eran Tromer, "Full AES key extraction in 65 milliseconds using cache attacks" presentation CRYPTO 2005 <http://www.iacr.org/conferences/crypto2005/rumpSchedule.html>

Additional resources may be added as research is done.